COMP 1204 UNIX Coursework Report

Name: Palak Jain

Student ID: 30012627

11/03/2019

The tar file was extracted by passing this command in the commandline:
`tar -zxvf reviews_dataset.tar.gz`

1 Scripts

1.1 Countreviews.sh

This script counts the number of reviews each hotel in a given directory gets and rearranges the hotels and the number of reviews such that the hotel with the most number of reviews is printed first (in descending order).

Here is a run through of how the script works: First, it checks whether or not a directory is given by the user through the command line. If it is not provided, the script skips to the else statement and the user is asked to input the relative or absolute path of a directory. If the directory is provided in the command line, a loop runs for all the files in the directory, the extensions from all filenames are removed and then their paths are removed. These new filenames are stored in the variable simplefilename each time the loop runs. The script then looks for the word 'Author' in each file and stores the number of times it appears in each file in the variable count. Finally, simplefilename and count are printed for each hotel file and they are arranged so that count is printed in descending order.

```
1  #!/bin/bash
2  #telling the environment/os to use bash as a command interpreter
3
4  #if an argument is given along with the script in the commandline
5  if [ $# -eq 1 ] && [ -d "$1" ]; then
6      #for every file in the directory taken from commandline
7      for file in $1/*; do
8          simplefilename=${file%.*} #remove extension
9          simplefilename=${simplefilename##*/} #remove path
10
11         #variable count stores the number of times
12         #<Author> field appears in each file
13         count=$(awk -F '>' '/<Author>/{sum +=1}
14
15         END{print sum}' $file)
16         echo $simplefilename $count
17
18     #sort reviews count in descending order
19     done | sort -k2 -n --reverse
20 else
21     #if absolute/relative path is not provided in the
22     #command line,inform the user that it must be provided
23     echo "Please provide the path for `basename $0`"
24 fi
25 #EOF
```

Listing 1: countreviews.sh

1.2 Averagereviews.sh

This script finds the average ratings each hotel entered in Trip Advisor gets from reviews' authors and orders them such that the hotel with most ratings appears first.

The script for this section is very similar to countreviews script. However, instead of the number of reviews in each file, we are interested in the average ratings given to each hotel. Therefore, we identify the Overall field, which gives us the ratings given by each author. The Overall field identifies as the first column and the actual ratings, as the second. So we find the sum of column 2 and find the average by dividing the sum by number of Overall fields in each file. By doing this, we end up getting the average of the ratings. The average of the ratings are then printed to 2 decimal places.

```
#!/bin/bash
#telling the environment/os to use bash as a command interpreter

#if an argument is given along with the script in the commandline
if [ $# -eq 1 ] && [ -d "$1" ]; then
    for file in $1/*; do #for every file in the given folder
        simplefilename=${file%.*} #remove extension
        simplefilename=${simplefilename##*/} #remove path

        #variable avg stores the result of the sum of all
        #the overall ratings divided by the total number of
        #times the field <Overall> appears
        avg=$(awk -F '>' '/<Overall>/{sum += $2; total += 1}

        #the result is limited to 2 decimal places
        END{printf ("%.2f", sum/total)}' $file)
        echo $simplefilename $avg

    #sort reviews count in descending order
    done | sort -k2 -n --reverse
else
    #if absolute/relative path is not provided in the
    #command line,inform the user that it must be provided
    echo "Please provide the path for `basename $0`"
fi
#EOF
```

Listing 2: averagereviews.sh

2 Discussion

2.1 Compare the use of unstructured markup vs. structured database for representing the data.

Structured markup is written in a format that is easy for machines to understand. As a result, in this format, data is easily searchable by basic algorithms. Unstructured data, on the other hand, makes it very difficult for algorithms to

search for specific data. Unstructured data can be useful in the way that it can give more information than one is looking for. However, if unstructured markup is altered slightly, it can be difficult to search for specific data, and one can also end up retrieving incorrect data.

2.2 Present 2-3 ideas to authenticate the authors of reviews.

Authors of reviews can be asked to provide evidence along with their reviews to solidify their experiences. Examples of evidence include images or email exchanges. In addition, Trip Advisor could develop an algorithm to identify authors who may have entered many reviews for one hotel using different author names, or a program to identify authors who perhaps wrote similar negative reviews for a particular hotel's main rivals. Both these cases could result as a tactic for hotels to get better reviews/ look better amongst its competitors.

2.3 Present 2-3 ideas on improving the review ranking system.

The review ranking system is not very reliable for a number of reasons. First of all, the number of reviews for each hotel varies. For example, new businesses could have recently joined the site and received a flurry of good reviews, resulting in them skyrocketing in the rankings.In turn, this would impact well-established companies that had been around for longer and had more reviews. Trip Advisor is in fact developing an algorithm that stabilizes the ranking of these new businesses. Another problem that could arise is that as time progresses, reviews get less reliable as hotels may go through changes overtime. Trip advisor could develop an algorithm to detect old reviews and remove them if they are no longer valid. In addition, some reviews are very poorly delivered and add very less for people seeking reviews for hotel advice. Trip advisor can develop a machine-learning algorithm which detects these kind of reviews and removes them from the database.

2.4 Discuss data storage issues with Trip Advisor's flat file-structure.

A flat file database like that of Trip Advisor's, is a database that stores data in a plain text file. Each line of the text file holds one record, with fields separated by delimiters, such as commas or tabs. A flat file database cannot contain multiple tables like a relational database can, and as a result, a large database, like that of Trip Advisor will most likely have data which is unnecessarily repeated several times in the same table. This results in an increased file sizes for the files in the database. The database only serves as a means of storing table information, but do not hold relations between the tables included. As a result, making changes to files can lead to errors. Such structure is supported mostly because of the ease with which it can carry data from the server, when no data

manipulation is expected and data is only meant to be read, stored and sent. This is a significant problem for TripAdvisor, as they would want to update their data on a regular basis.